

MULTI-OBJECTIVE HYBRID OPTIMAL CONTROL FOR MULTIPLE-FLYBY LOW-THRUST MISSION DESIGN

Jacob A. Englander*, Matthew A. Vavrina†, Alexander R. Ghosh‡

Preliminary design of low-thrust interplanetary missions is a highly complex process. The mission designer must choose discrete parameters such as the number of flybys, the bodies at which those flybys are performed, and in some cases the final destination. In addition, a time-history of control variables must be chosen that defines the trajectory. There are often many thousands, if not millions, of possible trajectories to be evaluated. The customer who commissions a trajectory design is not usually interested in a point solution, but rather the exploration of the trade space of trajectories between several different objective functions. This can be a very expensive process in terms of the number of human analyst hours required. An automated approach is therefore very desirable. This work presents such an approach by posing the mission design problem as a multi-objective hybrid optimal control problem. The method is demonstrated on a hypothetical mission to the main asteroid belt.

INTRODUCTION

Preliminary design of low-thrust interplanetary missions is a highly complex process. The designer must choose the launch date, flight time, propulsive maneuvers, and possibly a sequence of planetary flybys as well as altitudes and velocity vectors for each of those flybys. For some types of missions, such as missions to asteroids and comets, the designer is also responsible for choosing the destination because the customer is interested in a population of bodies rather than a specific body. Low-thrust missions add an additional degree of complexity because the designer must also choose a time history of control variables, *i.e.* thrust magnitude and direction, which define the trajectory. Furthermore a mission designer does not work in isolation - the customer who commissions the work, usually a scientist, does not just want a point solution even if that solution is globally optimal in propellant use, time, or some other metric. Rather, an exploration of the trade space between several metrics of the customer's choice is the true goal of preliminary mission design.

The traditional method of preliminary design is to approximate the low-thrust trajectory with a low-fidelity method such as a ballistic arc [1] or a shape-based approximation [2, 3, 4] and then to create a low-fidelity multiple-flyby trajectory by linking together many such arcs. A grid search tool [1] or a graphical tool combined with the designer's intuition [5] are then used to find the best low-fidelity solution. This solution is in turn used as an initial guess for a gradient-based low-thrust optimizer. Several such tools exist, including Mission Analysis Low-Thrust Optimization (MALTO) [6] and Gravity Assisted Low-thrust Local Optimization Program (GALLOP) [7]. This approach suffers from two significant drawbacks. First, the gradient-based optimizers converge to the locally optimal solution in the neighborhood of the initial guess and there is no guarantee that the best ballistic trajectory is in the neighborhood of the best low-thrust trajectory. In fact, the optimal flyby sequence or even the optimal target selection for a small body mission may not be the same

* Aerospace Engineer, Navigation and Mission Design Branch, NASA Goddard Space Flight Center, Greenbelt, MD, 20771, USA, Member AIAA

† Senior Systems Engineer, a.i. solutions, Inc., 10001 Derekwood Ln. Suite 215, Lanham, MD, 20706, USA

‡ Postdoctoral Fellow, Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, 104 S Wright Street, Champaign, IL, 61820, USA

between the best impulsive and low-thrust solutions. Second, it is computationally expensive to evaluate every possible ballistic solution in a grid and expensive in terms of human analyst time to try every promising candidate in a low-thrust solver. A more automated method that intelligently explores the design space is highly desirable. Such a method should be capable of finding solutions (a) with less cost, in terms of human analyst time, and (b) that may not be easily findable with a grid search or intuitive approach.

One such automated approach is to formulate the interplanetary design problem as a hybrid optimal control problem (HOCP). A HOCP is an optimization problem that is composed of two separable sub-problems, one with discrete variables and the other with continuous variables [8, 9]. For interplanetary design, the first problem is to choose the discrete parameters that define the mission, such as number of flybys, choice of flyby bodies, and, for some types of missions, the destination. The second problem is to find the time history of control variables, such as launch date, flight times, thrust magnitude and direction, flyby altitudes, and encounter velocity vectors that characterize the optimal trajectory for each set of discrete parameters. A HOCP can be solved using two nested optimization loops. The “outer-loop” solves the integer programming problem defining the discrete parameters. Each candidate solution to the “outer-loop” problem defines an “inner-loop” trajectory optimization problem. This approach was demonstrated first by Chilan, Wall, and Conway [10] for trajectories without flybys and then by Englander, Conway, and Williams for trajectories that include flybys and either impulsive chemical propulsion [11] or low-thrust electric propulsion [12]. All of these methods used a genetic algorithm (GA) to solve the outer-loop problem and a variety of stochastic global search algorithms to solve the inner-loop problem.

Other researchers have addressed components of the outer-loop problem. Gad and Abdelkhalik [13, 14] solve the multiple-flyby problem with impulsive, chemical thrust by using a single GA optimization loop. Another method, by Vasile and Campagnola [15], uses a set of successive deterministic algorithms to find candidate low-thrust, multiple flyby trajectories.

However, all of the methods above find only a single “optimal” trajectory, that is, optimal according to a single objective function. Preliminary mission design requires the exploration of a multi-objective trade space. The designer must find not a single solution but instead the Pareto front, surface, or hyper-surface (depending on the number of objectives) between several objective functions. Several researchers have addressed such problems in the past for problems with a fixed flyby sequence and fixed destination. Coverstone-Carroll, Hartmann, and Mason [16] used a multi-objective GA with an indirect trajectory optimizer. Vavrina and Howell [17] also used a multi-objective GA hybridized with a direct trajectory optimization method. Both research groups found non-dominated fronts of delivered mass versus flight time. In addition, Vasile and Zuiani [18] demonstrated a multi-objective algorithm for finding the non-dominated front between flight time and Δv for impulsive-thrust missions with fixed destination and flyby sequence. These authors are not aware of any method which solves the full, coupled multi-objective optimization problem while selecting both the discrete sequence parameters that define the mission and the continuous control parameters that define the trajectory.

In this work we present a new framework for multi-objective optimization of low-thrust interplanetary trajectories where the flyby sequence, and sometimes the destinations themselves, are not known *a priori*. The mission design problem is formulated as a HOCP where the outer-loop chooses the number of flybys, the identity of the flyby bodies, and, when appropriate, the destination. The outer-loop is based on the “null-gene” transcription presented by Englander, Conway, and Williams [11], a “cap and optimize” approach for varying the flight time and launch date, and the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) multi-objective GA developed by Deb [19]. The inner-loop is based on the Sims-Flanagan transcription [20] combined with the monotonic basin hopping (MBH) global search algorithm [21, 22, 12, 23]. The method is demonstrated on a hypothetical mission to the main asteroid belt.

PHYSICAL MODELING

Mission Architecture

Three layers of event types are defined in this work: *missions*, *journeys*, and *phases*. A *mission* is a top-level container that encompasses all of the events including departures, arrivals, thrust arcs, coast arcs, and flybys. A *journey* is a set of events within a mission that begin and end target of interest, i.e. not just a body

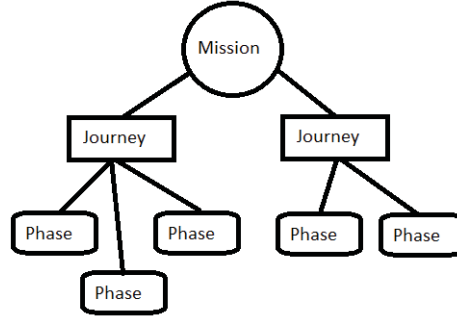


Figure 1: Anatomy of a Mission

that is being used for a propulsive flyby. For example, the interplanetary cruise portion of the Cassini mission was composed of a single journey that began at Earth and ended at Saturn. JAXA’s Hayabusa mission, which rendezvoused and took samples from near Earth asteroid Itokawa, had two journeys - one from Earth to Itokawa, and one from Itokawa to Earth. NASA’s Dawn mission is also composed of two journeys, one from Earth to Vesta and one from Vesta to Ceres. Each journey is composed of one or more *phases*. Like a journey, a phase begins at a planet and ends at a planet, but unlike the end points of a journey, the end points of a phase may represent a flyby of a body that is being used only to modify the trajectory of the spacecraft, i.e. a propulsive flyby. For example, the first journey of the Dawn mission may be considered to be a two-phase journey because it included a flyby of Mars. The number of journeys in a mission is fixed *a priori* but the number of phases is not, and in the context of this work both the number of phases and the identity of the flyby planets in each phase may be chosen by the optimizer. Figure 1 is a block diagram of a mission using the journey/phase nomenclature.

The Sims-Flanagan Transcription

The Sims-Flanagan transcription is a widely used method in which the continuous-thrust trajectory is discretized into many small time steps, and the thrust applied during each time step is approximated as a small impulse placed at the center of the time step. The trajectory is propagated between control points by solving Kepler’s problem [20]. The Sims-Flanagan transcription, when used with a nonlinear programming (NLP) solver such as Sparse Nonlinear Optimizer (SNOPT) and a suitable initial guess, is very fast and robust. It is considered to be a “medium-fidelity” transcription and is used in existing software packages such as MALTO [6], GALLOP [7], and Parallel Global Multiobjective Optimizer (PaGMO) [24].

In the classical Sims-Flanagan transcription, the optimizer chooses the three components of an impulsive $\Delta \mathbf{v}$ vector at the center of each time-step. In order to improve the robustness of the solver, a modified transcription known as “up-to-unit vector control” is used in the Evolutionary Mission Trajectory Generator (EMTG), where instead of choosing the $\Delta \mathbf{v}$ vector directly the optimizer instead chooses a control 3-vector in $[-1.0, 1.0]$ that is multiplied by the maximum Δv that the spacecraft can produce in that time-step. The magnitude of the control vector is bounded in the range $[0.0, 1.0]$, i.e.,

$$\Delta \mathbf{v}_i = \mathbf{u}_i \Delta v_{max,i}, \|\mathbf{u}_i\| \leq 1.0 \quad (1)$$

where

$$\Delta v_{max,i} = \frac{D n_{available} T_{max} (t_f - t_0)}{mN} \quad (2)$$

where D is the thruster duty cycle, $n_{available}$ is the number of available thrusters, T_{max} is the maximum available thrust from one thruster, t_0 and t_f are the beginning and ending times of the time step, m is the mass of the spacecraft at the center of the time step, and N is the number of time steps in the phase. This modified Sims-Flanagan transcription is used in MALTO, Parallel Global Multiobjective Optimizer (PaGMO), and EMTG.

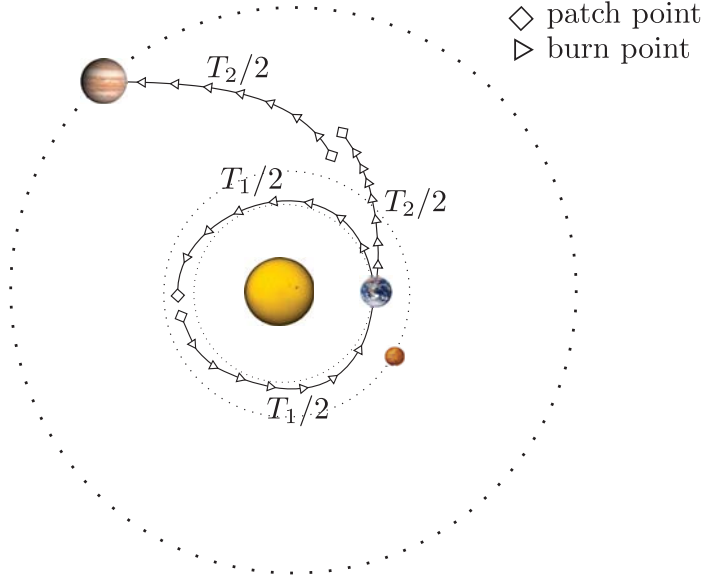


Figure 2: An Example Trajectory Using the Sims-Flanagan Transcription

The spacecraft state is propagated forward from the first endpoint (i.e. planet) in each phase and backward from the second endpoint. The trajectory is propagated by solving Kepler's equation and the spacecraft mass is propagated by assuming a constant mass flow rate across the each time-step. The specific Kepler propagator algorithm used in EMTG is a Laguerre-Conway method [25, 26]. A set of nonlinear constraints are applied to ensure continuity in the center of the phase,

$$\mathbf{s}_{mf} - \mathbf{s}_{mb} = \begin{bmatrix} \Delta x & \Delta y & \Delta z & \Delta v_x & \Delta v_y & \Delta v_z & \Delta m \end{bmatrix} = \mathbf{0} \quad (3)$$

The optimizer also chooses the initial and final velocity vectors for each phase. If a phase begins with a launch, the magnitude of the initial velocity vector is used with a launch vehicle model to determine the initial mass of the spacecraft as described later in this work. If a phase begins with a planetary flyby, two nonlinear constraints are applied to ensure that the flyby is feasible. First, the incoming and outgoing velocity vectors with respect to the planet must be equal,

$$\mathbf{v}_{\infty}^+ - \mathbf{v}_{\infty}^- = \mathbf{0} \quad (4)$$

where \mathbf{v}_{∞}^- and \mathbf{v}_{∞}^+ are the velocities before and after the flyby, respectively. Second, the spacecraft may not fly closer to the planet than some user-specified minimum flyby distance:

$$\frac{\mu_{planet}}{v_{\infty}^2} \left[\frac{1}{\sin(\frac{\delta}{2})} - 1 \right] - (r_{planet} + h_{safe}) \geq 0 \quad (5)$$

where

$$\delta = \arccos \left[\frac{\mathbf{v}_{\infty}^- \cdot \mathbf{v}_{\infty}^+}{(v_{\infty}^-)^2 (v_{\infty}^+)^2} \right] \quad (6)$$

Here μ_{planet} is the gravitational parameter of the planet, r_{planet} is the radius of the planet, δ is the flyby turn angle, and h_{safe} is the user-defined minimum altitude.

Figure 2 is a diagram of a simple low-thrust mission to Jupiter with one Earth flyby using the multiple gravity assist with low-thrust (MGALT) model. The continuity constraints are deliberately left unsatisfied in the diagram to illustrate where they must be applied.

There are four significant advantages to using the Sims-Flanagan transcription. First, the optimal objective function value for a Sims-Flanagan trajectory design is usually very close to the optimal cost value for a

higher-fidelity version of the same trajectory. Second, a low-thrust trajectory generated using the Sims-Flanagan transcription makes a very good initial guess for a higher-fidelity trajectory design. Third, the Sims-Flanagan transcription is very fast because it does not require numerical integration of differential equations. Fourth, the convergence of an NLP solver solving a Sims-Flanagan problem is very robust to poor initial guesses, making it ideal for an automated design approach.

Launch Vehicle, Propulsion, Power, and Ephemeris Modeling

Low-thrust trajectories are inextricably coupled to the specific hardware used by the spacecraft. The optimal trajectory for one combination of launch vehicle, propulsion system, and power system will not be the optimal trajectory for a different hardware combination. EMTG therefore employs realistic modeling of these three systems and a system for applying margin to launch vehicle, propulsion, and power system performance. These models are omitted from this paper for the sake of brevity but interested readers may find them in Reference [27]. In addition, the ephemerides of solar system bodies are provided using the high-fidelity SPICE ephemeris system [28], or, if SPICE kernels are unavailable, via static orbit elements.

OUTER-LOOP OPTIMIZATION OF THE MISSION SEQUENCE

Outer-Loop Transcription

The mission design problem in this work is posed as two nested optimization problems, an “outer-loop” discrete optimization problem and an “inner-loop” real-valued optimization problem. The outer-loop solves a multi-objective integer programming problem whose candidate solutions are themselves instances of the inner-loop trajectory optimization problem. The outer-loop works via a “cap and optimize” process by which it chooses design variables such as destinations, flybys, and bounds on the launch date and flight time which define a tractable inner-loop subproblem.

The user specifies *a priori* a list of outer-loop design variables and a “menu” of choices with corresponding integer codes for each. In this work the design variables are launch epoch, time of flight, destinations, and flybys. The outer-loop algorithm makes one choice from each menu.

Launch epoch is transcribed as a menu of candidate launch dates plus a launch window size. For example, the user might specify launches in 2020, 2021, or 2022 with a 365 day launch window. The outer-loop chooses one of the available launch years and constructs an inner-loop problem where the spacecraft can depart Earth at any time during that year. The method for time of flight is similar - the user specifies a list of flight times and the optimizer chooses one and sets it as the upper-bound for the inner-loop problem. No lower-bound is enforced - this seems to yield a more tractable inner-loop problem.

The user may also specify a menu of candidate destinations for each journey. For example, one might wish to design a mission to two asteroids but have a long list of scientifically interesting options. The HOCF automaton can choose the most accessible asteroids. The outer-loop may be instructed to discard candidate solutions that visit the same destination body more than once.

Flyby sequence selection is similar to journey destination selection except that one does not always know how many flybys are to be performed. A “null-gene” technique is used to choose the number and identity of flyby bodies [11]. The analyst provides a list of acceptable flyby bodies and a maximum number of flybys for each journey. Then, for each potential flyby, the outer-loop may select from a list containing the specified acceptable bodies and also a number of “null” options equal to the number of acceptable bodies. The outer-loop therefore has an equal probability of selecting “no flyby” for each opportunity as it does to select a flyby. This technique has been shown to be very effective for designing multi-flyby interplanetary missions and has been used to reproduce the Cassini [11] trajectory and design an efficient variant of the BepiColombo trajectory [29].

The user may then select any number of outer-loop objective functions for optimization. Some of these, such as flight time and launch year, may be directly related to decision variables. Others, such as final mass, may be the product of an the inner-loop optimization process. The inner-loop subproblem is optimized over

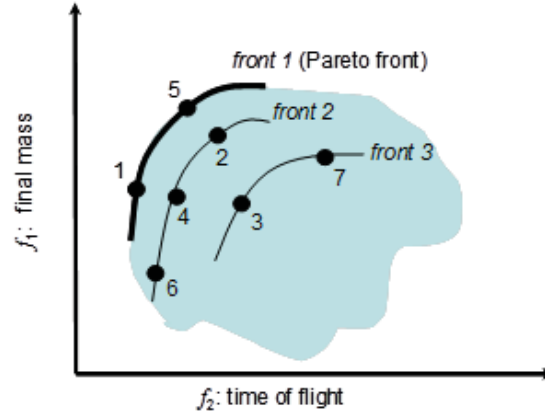


Figure 3: Two-Dimensional, Non-Dominated Fronts of a Multi-Objective Trajectory Optimization Example

only one objective function but because the “cap and optimize” outer-loop constrains the inner-loop, each sub-problem solution is a candidate solution to the multi-objective outer-loop problem.

Outer-Loop Multi-Objective Optimization via NSGA-II

The goal of multi-objective optimization is to generate the Pareto front of solutions [30]. The Pareto front represents the set trade-off solutions in which no improvement can be achieved in one objective without degrading at least one other objective. That is, all designs that compose the Pareto front are equally optimal. The Pareto front can be discontinuous and either concave or convex. Thus, the aim of multi-objective optimization is to generate numerous Pareto-optimal solutions such that a representation the Pareto front is created to enable a tradeoff decision. The Pareto front for a notional low-thrust trajectory optimization problem with the objectives to maximize final spacecraft mass and minimize time of flight is illustrated in Figure 3. The multi-objective optimization problem can be stated as:

$$\begin{aligned}
 &\text{minimize } \mathbf{f}(\mathbf{x}) \\
 &\text{subject to } \mathbf{c}(\mathbf{x}) \leq \mathbf{0} \\
 &\quad \mathbf{x}_{LB} \leq \mathbf{x} \leq \mathbf{x}_{UB}
 \end{aligned} \tag{7}$$

where $\mathbf{f}(\mathbf{x})$ is a vector of objective functions, \mathbf{x} is a vector of design variables, $\mathbf{c}(\mathbf{x})$ is a vector of constraint functions, and \mathbf{x}_{LB} and \mathbf{x}_{UB} are vectors of upper and lower bounds for \mathbf{x} , respectively. The objective functions are often coupled, i.e. contain the same design variables, and also competing, i.e. the optimal solution with respect to one objective is not also the optimal solution with respect to the other objectives. Competition between objectives creates the need to find multiple solutions, making multi-objective optimization more complex than single-objective optimization. The outer-loop of this work uses bounded but unconstrained multi-objective optimization, with the constraints \mathbf{c} handled by the single-objective inner-loop.

The multi-objective optimization concept of domination allows for the comparison of a set of designs with multiple objectives, providing a measure of the relative quality of the design. When comparing two multi-objective designs, the design \mathbf{x}_1 dominates design \mathbf{x}_2 if:

$$\begin{aligned}
 &\forall p : f_p(\mathbf{x}_1) \leq f_p(\mathbf{x}_2) \quad \text{where } p = 1, 2, \dots, n_{\text{objectives}} \\
 &\exists p : f_p(\mathbf{x}_1) < f_p(\mathbf{x}_2)
 \end{aligned} \tag{8}$$

That is, \mathbf{x}_1 dominates design \mathbf{x}_2 if, for all objectives p , \mathbf{x}_1 is better than or equal to \mathbf{x}_2 , and \mathbf{x}_1 outperforms \mathbf{x}_2 for at least one objective. In a direct comparison of two designs, if one design dominates another, that design is closer in proximity to the Pareto front. If neither design dominates the other, the designs are non-dominant to each other. Therefore, in a set of designs, the superior designs are those that are not dominated

by any other design in the set, and are termed the non-dominated subset. It follows that any Pareto-optimal design is a member of the non-dominated subset associated with the entire feasible objective space and is located along the Pareto front. A solution space with several Pareto fronts is shown in Figure 3.

A multi-objective genetic algorithm (MOGA) is an appropriate solution method for the outer-loop problem. A GA is a stochastic search and optimization technique that simulates natural selection and reproduction with the goal of identifying globally-optimal designs. A GA first generates a random population of designs and then iteratively executes three genetic operators: selection, crossover, and mutation. The operators are applied to a parent population to produce a new offspring generation that is better adapted to fitness landscape defined by the objective functions. A GA does not require an initial guess or gradient information, making it ideal for exploring discrete, multi-modal, and expansive design spaces. A MOGA is a modified version of the standard GA which can solve multi-objective problems.

A MOGA is capable of finding the Pareto front for a multi-objective problem in a single optimization run. This is more efficient, in terms of computer time required, than performing many repetitions of a single-objective optimization routine. One effective MOGA is the Non-Dominated Sorting Genetic Algorithm (NSGA) developed by Deb [31], in which the fitness of an individual in the population is based on its relative proximity to the population's non-dominated front. The genetic operators of the NSGA evolve the population toward the globally-optimal Pareto front in the same way that the population of a single-objective GA evolves toward the globally optimal solution.

A second generation non-dominated sorting genetic algorithm, the NSGA-II, improves upon the original NSGA [19]. The NSGA-II incorporates mechanisms that ensure that the elite individuals, i.e. the best individuals in the population, are retained as the population evolves. Additionally, the NSGA-II employs strategies that aim to produce a uniform representation of designs along the Pareto front. Domination is used to categorize each design into non-dominated fronts. The solutions composing the best non-dominated front are given a rank of one, and each subsequent front is given an incrementally higher rank according to their relative distance to the Pareto front. The fast non-dominated sorting algorithm by Deb is outlined in Algorithm 1, where m and n are individual solutions, S_m is the set of solutions which dominate m , c_m is the number of solutions which dominate m , F_i are sets which contain the non-dominated fronts, Q is a temporary set, and n_{rank} is rank of solution n .

The NSGA-II preserves diversity by preferring the solution that is in a less dense region of the objective function space when comparing two solutions of the same rank. A crowding distance is assigned to each individual based on the perimeter of the hyper-rectangle formed with the two adjacent designs in objective space for each of each objective. The HOCP automaton described in this work employs a modified version of Deb's [19] crowding distance assignment to accommodate instances in which there are more than two objectives and there are multiple solutions belonging to the same non-dominated front with the same minimum objective function value. Whenever there are multiple solutions that take the minimum or maximum value for a given objective function coinciding in the same non-dominated front, the solutions with the lowest objective function value for the other objectives are given a large crowding distance. The remaining solutions in the boundary group are assigned a crowding-distance based on a single adjacent solution for each objective. This approach ensures that solutions with a minimum objective function value will not be discarded if the entire population composes a single non-dominated front. The modified crowding distance assignment operator is described in Algorithm 2, where I is the set of solutions to be sorted and the operator $sort(I, p)$ refers to sorting a list I in ascending order by objective p . L is the set of solutions for a given objective p for which p is at a minimum and H is the set of solutions for which p is at a maximum, at the current generation.

After ranking the population using the non-dominated sort and crowding distance algorithms, NSGA-II applies genetic operators to the parent population to create an offspring population of the same size N . The parent and offspring populations are then combined into a single population of size $2N$. This combined population is then sorted and ranked according to non-domination. The next generation's parent population is created by taking each non-dominated front from the combined population, in rank order, until the new parent population is filled. In this way the preservation of elite individuals is guaranteed and diversity in objective space is promoted. The process is repeated for a set number of generations or length of time.

Algorithm 1 Fast non-dominated sort [19]

```
for each individual  $m$  in population  $P$  do
   $S_m = \emptyset$ 
   $c_m = 0$ 
  for each individual  $n$  in population  $P$  do
    if  $m$  dominates  $n$  then
       $S_m = S_m \cup \{n\}$ 
    else if  $n$  dominates  $m$  then
       $c_m = c_m + 1$ 
    end if
  end for
  if  $c_m = 0$  then
     $m_{rank} = 1$ 
     $F_1 = F_1 \cup \{m\}$ 
  end if
end for
 $i = 1$ 
while  $F_i \neq \emptyset$  do
   $Q = \emptyset$ 
  for each individual  $m$  in front  $F_i$  do
    for each individual  $n$  in  $S_m$  do
       $c_m = c_m - 1$ 
      if  $c_m = 0$  then
         $n_{rank} = i + 1$ 
         $Q = Q \cup \{n\}$ 
      end if
    end for
  end for
   $i = i + 1$ 
   $F_i = Q$ 
end while
```

The NSGA-II is well-suited for the multi-objective low-thrust trajectory optimization problem: it is capable of generating the Pareto front to illustrate the trade-offs in the mission objectives of interest, can globally search the design, and is automated without requiring an initial guess.

Parallel Outer-Loop Optimization

The evaluation of the objective functions for a candidate solution is very expensive - it requires solving the entire inner-loop trajectory optimization problem and can take at best several seconds, in most cases many minutes, and in some cases hours. Fortunately each inner-loop subproblem is independent of the others so it is natural to evaluate them in parallel. The pool of N inner-loop subproblems to be evaluated is distributed over P processor cores. If N exceeds P , the subproblems are agglomerated into P task pools with one assigned to each processor. The run-time is further decreased by saving each candidate solution to the outer-loop problem so that none must be evaluated more than once. Therefore the number of inner-loop instances to be run for each outer-loop generation tends to decrease and the algorithm speeds up with each generation of NSGA-II.

There are two parallel processing paradigms: shared memory multiprocessing and distributed memory multicomputing. Shared memory multiprocessing, also known as threading, is easier to implement but is not appropriate for this work because it requires all processes to be “thread-safe,” i.e. do not interfere with each other even while operating in the same memory space. However the SPICE ephemeris reader used in this work is not thread-safe and therefore the entire algorithm cannot be used with shared memory multiprocessing.

Algorithm 2 Modified Crowding-Distance Assignment

```
for each objective  $p$  do
   $I_p = \text{sort}(I, p)$ 
   $L = \emptyset$ 
   $H = \emptyset$ 
  for  $i$  in  $I_p$  do
    if  $p = 0$  then
       $I_p[i]_{\text{distance}} = 0$ 
    end if
    if  $f_p(I_p[i]) = f_p^{\min}$  then
       $A \cup B$ 
       $L = L \cup \{I_p[i]\}$ 
    else if  $I_p[i]_{\text{objective-value}} = f_p^{\max}$  then
       $H = H \cup \{I_p[i]\}$ 
    else
       $I_p[i]_{\text{distance}} = I_p[i]_{\text{distance}} + (f_p(I_p[i+1]) - f_p(I_p[i-1])) / (f_p^{\max} - f_p^{\min})$ 
    end if
  end for
  for each objective  $q$  except  $p$  do
     $L_q = \text{sort}(L, q)$ 
     $L_q[0]_{\text{distance}} = \infty$ 
     $H_q = \text{sort}(L, q)$ 
     $H_q[0]_{\text{distance}} = \infty$ 
  end for
end for
```

Instead the algorithm described in this work is implemented using the distributed memory multicomputing library message passing interface (MPI) [32] in which each processor runs an independent process in distinct memory spaces.

INNER-LOOP TRAJECTORY OPTIMIZATION

Stochastic Global Search via Monotonic Basin Hopping and SNOPT

Because the solutions to the outer-loop problem require solutions to the inner-loop trajectory optimization subproblem, the performance of the HOCF automaton is only as good as the performance of the inner-loop solver. A solver is required that can solve large, multi-modal problems with many nonlinear constraints. Because the inner-loop problem is generated in real time by the outer-loop, the inner-loop solver cannot require any human intervention and therefore cannot require an initial guess - a natural application for a heuristic stochastic search. The HOCF automaton described in this work uses monotonic basin hopping (MBH) [21, 22, 33, 23]. MBH, described in Algorithm 3, is a hybrid of a stochastic search step with an NLP solver. The stochastic search step efficiently explores the space and the NLP step enforces the constraints and exploits local basins. The NLP step in this work is performed using SNOPT [34].

MBH requires a vector of lower and upper bounds on each of the inner-loop decision vectors. These can be specified *a priori* but in the case of an HOCF automaton there is no opportunity for a human to intervene and set bounds. Therefore in this work a set of simple laws are used to determine the bounds. In the interest of brevity the table of bounds-choosing laws is omitted from this paper but may be found in [23].

The MBH+NLP optimization algorithm in EMTG is efficient and does not require an initial guess. MBH is most useful when one does not have much *a priori* information about the solution as is *always* the case when solving an inner-loop trajectory optimization problem inside a hybrid optimal control (HOC) mission design problem.

Algorithm 3 Monotonic Basin Hopping (MBH)

```
generate random point  $\mathbf{x}$ 
run NLP solver to find point  $\mathbf{x}^*$  using initial guess  $\mathbf{x}$ 
 $\mathbf{x}_{current} = \mathbf{x}^*$ 
if  $\mathbf{x}^*$  is a feasible point then
    save  $\mathbf{x}^*$  to archive
end if
while not hit stop criterion do
    generate  $\mathbf{x}'$  by randomly perturbing  $\mathbf{x}_{current}$ 
    for each time of flight variable  $t_i$  in  $\mathbf{x}'$  do
        if  $\text{rand}(0, 1) < \rho_{time-hop}$  then
            shift  $t_i$  forward or backward one synodic period
        end if
    end for
    run NLP solver to find point  $\mathbf{x}^*$  from  $\mathbf{x}'$ 
    if  $\mathbf{x}^*$  is feasible and  $f(\mathbf{x}^*) < f(\mathbf{x}_{current})$  then
         $\mathbf{x}_{current} = \mathbf{x}^*$ 
        save  $\mathbf{x}^*$  to archive
    else if  $\mathbf{x}^*$  is infeasible and  $\|c(\mathbf{x}^*)\| < \|c(\mathbf{x}_{current})\|$  then
         $\mathbf{x}_{current} = \mathbf{x}^*$ 
    end if
end while
return best  $\mathbf{x}^*$  in archive
```

EXAMPLES

The HOCF automaton applied in this work is applied to a challenging mission design problem in which the spacecraft is required to visit two main-belt asteroids that are over 50 km in diameter and are sufficiently well characterized to be classified in the Tholen taxonomy. There are 475 such bodies, and therefore 225150 combinations of bodies when one observes that visiting the same two asteroids in a different order defines a different mission and missions visiting the same asteroid twice are removed. It is possible to reduce the size of the decision space by pre-pruning the list of bodies by eccentricity, semi-major axis, or inclination, but for the purposes of this study the list was deliberately left unpruned to demonstrate the ability of the HOCF automaton to explore such a decision space.

The HOCF automaton chooses the launch year between 2020 and 2029 and the time of flight between 5 and 12 years. In addition flybys of the Earth, Mars, and Jupiter are permitted. The HOCF automaton may choose up to two flybys between launch and the first asteroid, plus up to one flyby between asteroids.

The setup of the problem is divided into two components: a list of assumptions applied to all candidate missions, show in Table 1, and the set of menus of choices for each outer-loop decision variable, shown in Tables 2 along with their corresponding integer codes. The duty cycle and power margin assumptions listed in Table 1 are standard in preliminary design at NASA Goddard Space Flight Center. Table 3 lists the settings applied to the NSGA-II outer-loop solver and Table 4 lists the settings applied to the MBH+NLP inner-loop solver.

Tables 2 combine to allow 4.82×10^9 possible missions, or 1.16×10^9 possibilities once duplicates are removed. It is impractical to evaluate all of the options via a grid search, so this problem is a natural fit for the HOCF automaton.

The HOCF automaton was run for 29 days on a 64-core cluster employing Scientific Linux. NSGA-II evaluated 286 generations in that time. Normally such a long run is not required but given the size of the decision space and the desire for statistically meaningful results, an extended run time was desirable. The outer-loop begins by evaluating a random population of mission candidates as shown in Figure 4. Note

Table 1: Assumptions Applied to All Candidate Missions

Description	Value
Number of Time-Steps per Phase	20
Launch Vehicle	Atlas V 401
Forced Post-Launch Coast	30 days
Forced Pre-and-post-Flyby Coast	15 days
Solar Array Performance at 1 AU	20.0
Solar Array Performance Model	$1/r^2$
Spacecraft Bus Power Model	constant 0.8 kW
Thruster	NEXT [35]
Thruster Duty cycle	90%
Propulsion Power Margin	15%

Table 2: Menus for Outer-Loop Decision Parameters

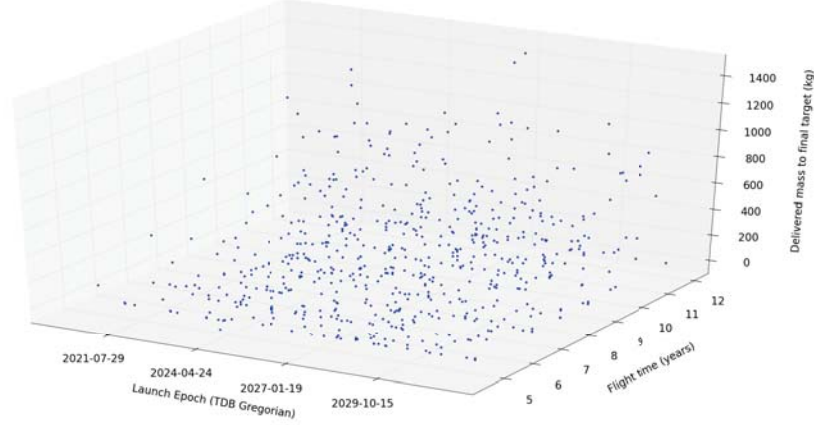
(a) Journey Destination (twice)		(b) Flyby Choices (three times)	
Code	Body	Code	Planet
0	Ceres	0	Earth
1	Pallas	1	Mars
2	Juno	2	Jupiter
3	Vesta	3	no flyby
...	...	4	no flyby
(475 choices)		5	no flyby
(c) Launch Year		(d) Flight Time	
Code	Year	Code	Flight Time (years)
0	2020	0	5
1	2021	1	6
2	2022	2	7
3	2023	3	8
4	2024	4	9
5	2025	5	10
6	2026	6	11
7	2027	7	12
8	2028		
9	2029		

Table 3: Outer-Loop NSGA-II Settings

Description	Value
Population Size	1024
Mutation Ratio	0.15
Objective Functions	launch epoch time of flight delivered mass to the second asteroid

Table 4: Inner-Loop MBH+NLP Settings

Description	Value
MBH step distribution [23]	Pareto
Pareto α [23]	1.5
Probability of MBH time hop $\rho_{time-hop}$ [33]	0.05
Maximum run-time for MBH	20 minutes
Maximum run-time for NLP	1 minute
Maximum number of iterations for NLP	8000
NLP feasibility tolerance	1.0×10^{-5}
Objective Function	maximize final mass

**Figure 4: Initial Population of Candidate Missions for the Multi Main-Belt Asteroid Tour Problem**

that the launch epoch, flight time, and delivered mass of the candidate missions in the initial population appear randomly distributed, showing that for this problem a random selection of variables in the outer-loop decision space yields a random distribution of values in the objective space. It is clear that many, if not all, of the missions in the initial population do not lie on the non-dominated surface for this problem. Also, only 635 of the 1024 candidate missions evaluated in the first generation - only 62% - were accepted by the inner-loop. That is, only 62% of the inner-loop subproblems generated by the outer-loop were successfully evaluated by the MBH+NLP optimizer. Some of these missions were discarded because they visit the same asteroid twice, but most of them were simply infeasible.

By the 10th outer-loop generation, as shown in Figure 5, a weak relationship between flight time and delivered mass becomes apparent. In general, the longer the flight time the greater the deliverable mass. This is most clear in the left-hand side of Figure 5, representing the missions with the earliest launch epoch. Many of the candidate missions are clustered at the beginning of the launch window. This is mainly because the optimizer has not yet found significant incentive to launch later in order to achieve better flight time and/or delivered mass. In general the candidate missions in the 10th generation deliver more mass than those in the initial population. This can be seen by comparing Figures 4 and 5 and noting that in Figure 5 the upper limit of the mass axis is higher and there are many points near the top of the plot. Also note that by the 10th generation the full 1024 candidate solutions are represented in the plot because there are no more infeasible missions remaining in the population.

Figure 6 shows the 20th generation of the outer-loop. By this time the population is much more widely spread in the objective space. The trend between longer flight times and higher delivered mass is more obvious and the missions that are not on the non-dominated front, i.e. missions with inferior delivered mass

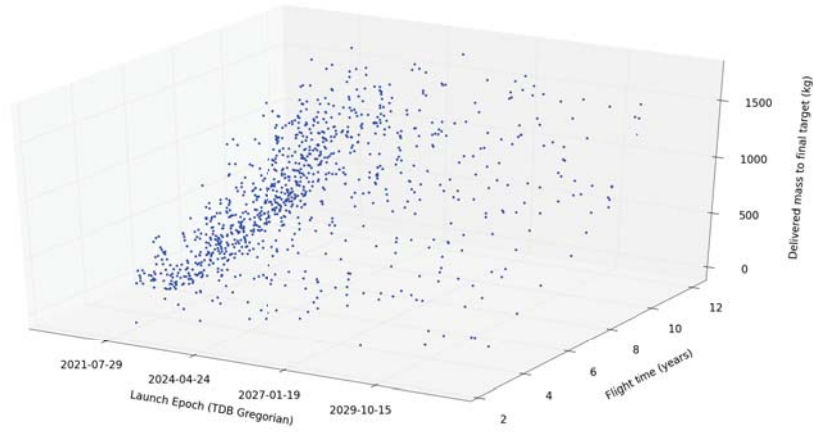


Figure 5: 10th Outer-Loop Generation for the Multi Main-Belt Asteroid Tour Problem

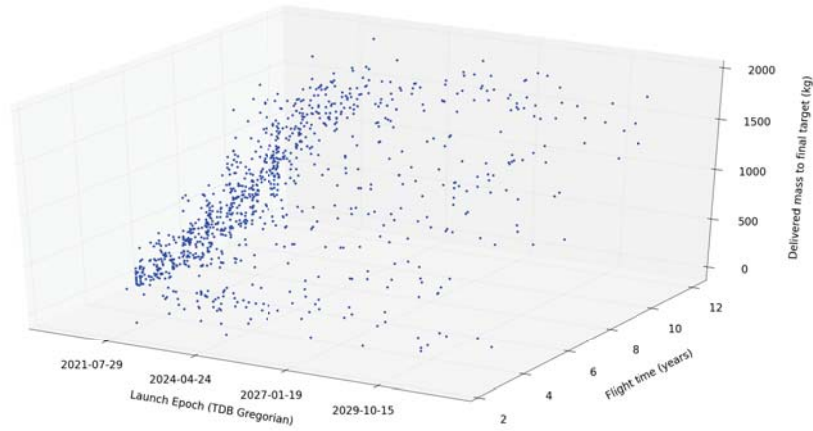


Figure 6: 20th Outer-Loop Generation for the Multi Main-Belt Asteroid Tour Problem

for a given launch epoch and flight time than other missions in the population, are starting to be pruned away. Also note that the upper limit of the mass axis has increased again and there are several missions that deliver nearly 2000 kg. There are still many points clustered on the left-hand side of the plot, showing that the outer-loop has not identified a strong relationship between launch epoch and the other two objectives.

By the 50th outer-loop generation, as shown in Figure 7, the line of points on the left-hand side of the plot has risen higher on the mass axis. This result means that the outer-loop is finding more and more high-mass solutions without having to sacrifice flight time. This is because the algorithm is finding asteroids that are easier to reach.

Figure 8 shows the 100th generation of the outer-loop. By generation 100 the maximum delivered mass is no longer increasing but outer-loop has more fully explored the short flight time region. Also, the population is in general higher on the mass axis, showing that the outer-loop is continuing to find better asteroid pairs. Finally the population has spread out on the launch epoch axis to find missions that depart Earth in the late 2020s.

The 287th and final generation is shown in Figure 9. This last generation is very similar to the 100th but has some new features. Most notably there is a new line of points on the left-hand side of the plot showing trajectories with a higher mass per unit flight time than before. Otherwise the plot shows that the population

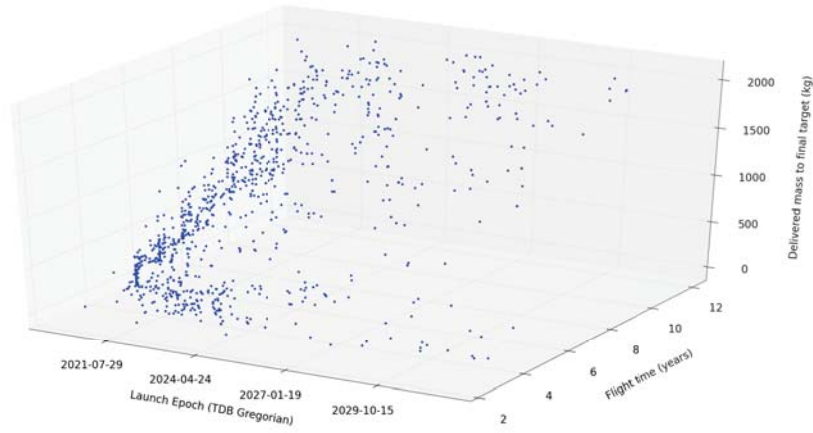


Figure 7: 50th Outer-Loop Generation for the Multi Main-Belt Asteroid Tour Problem

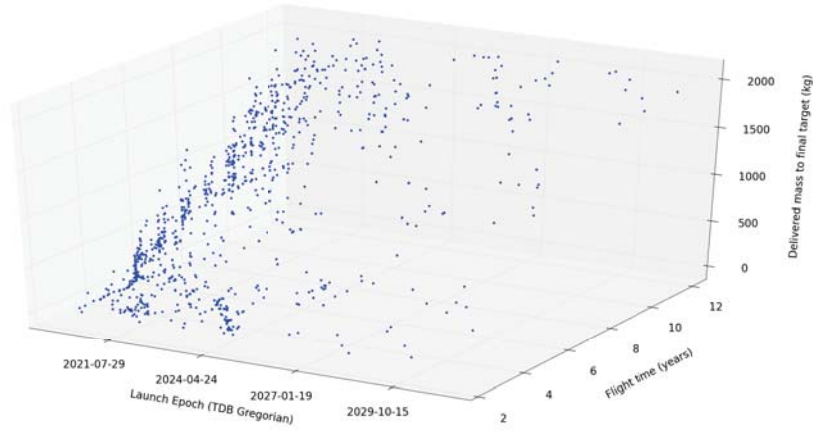


Figure 8: 100th Outer-Loop Generation for the Multi Main-Belt Asteroid Tour Problem

has not changed very much since the 100th generation. In order to prepare this paper for publication, the optimization was stopped at generation 287. However it is clear that there are still some points remaining on the plot that are not part of the non-dominated front, that is, they are dominated by some solutions that are on the front. If the outer-loop were run for several more tens of generations, those points might disappear. Unfortunately this was not practical due the paper deadline but will be done in future work.

One of the most interesting questions about methods based on multi-objective stochastic search heuristics, such as the outer-loop in this work, is how to measure convergence. Unfortunately there is no proof of global convergence. However one way to tell when an NSGA-II run is nearing completion, i.e. if its discovery of useful new solutions is slowing down, is to perform a statistical analysis of the age of the missions in the population. As the optimization process proceeds, a record is kept of every mission evaluated and in which generation it first appeared, i.e. its “birth generation.” The mean birth generation of a mission in the final population is 158 and the median birth generation is 164, with a standard deviation of 79 generations. Since generation 287 is more than one standard deviation away from the mean birth generation, most of the “work” that is likely to be accomplished by the outer-loop GA is complete. One may reasonably ask if a final generation that is 1.63 standard deviations larger than the mean birth generation is sufficient. It may be possible, in future work, to determine a stopping condition for the outer-loop based on the mean birth generation and standard deviation but at this time we cannot draw a conclusion.

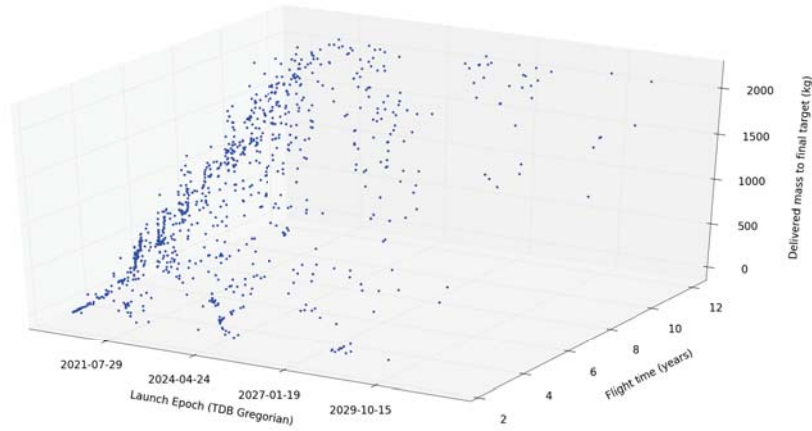


Figure 9: 287th Outer-Loop Generation for the Multi Main-Belt Asteroid Tour Problem

Of the 476 possible asteroid choices, 474 of them appear in feasible missions. 246 asteroids are represented in the final population of missions and represent a feasible subset. The most commonly chosen asteroid in the final population was 207 Hedda, but two other asteroids, 63 Ausonia and 172 Baucis, were also very popular in previous generations. Figure 10 shows the number of missions in each generation that included the three most popular asteroids. 63 Ausonia and 172 Baucis trade very well against the rest of the population in early generations and very quickly dominate the population, commonly as a pair in a single mission. 207 Hedda takes longer to come to prominence but by the 287th generation is the most common asteroid. One can conclude from these statistics that 207 Hedda, 63 Ausonia, and 172 Baucis are the most accessible asteroids in the trade space. No other asteroid appears nearly as many times as these three, with the next most common asteroid, 554 Peraga, peaking at 87 appearances in the 158th generation and then sliding back to only 49 appearances in the 287th generation.

It is instructive to examine a few missions from the final population in detail. First, suppose the customer, i.e. the scientist who has commissioned the study, is interested only in the subset of missions that have flight times no longer than 10 years and deliver no less than 1000 kg to the second asteroid. Figure 11 shows the population of missions in the 287th generation, zoomed in such that only the interesting subset is visible. Three example missions are circled in Figure 11: a short-duration mission (A) which launches in 2021 and delivers 1005 kg with a flight time of less than 6 years, a long-duration mission (B) which launches in 2023 and delivers 2055 kg with a flight time of ten years, and a medium-duration mission (C) which launches in 2029, near the end of the decade of interest, and delivers 1530 kg with a flight time of 8 years.

Example mission (A), shown in Figure 12, is an example of a fast mission which delivers the minimum amount of mass required for the mission, in this case 1000 kg. Example mission A travels directly to 207 Hedda and 20 Massalia, avoiding planetary flybys which can improve delivered mass at the cost of flight time. The grey lines in Figure 12 represent the orbits of the Earth and the asteroids. The dashed blue line represents the path of the spacecraft during the first 30 days after launch in which thrusting is not permitted because that time is reserved for checkout of the spacecraft. The solid black line represents the path of the spacecraft when the optimizer chooses to thrust, and the dashed black line represents the path of the spacecraft when the optimizer chooses to coast. The solid red lines represent the thrust vector at the center of each time-step.

Example mission (B), shown in Figure 13 is an example of a long-duration mission which delivers the largest possible mass at the cost of flight time. The spacecraft performs a flyby of the Earth one year after launch and then a second flyby of Mars one year later before arriving at 554 Peraga. There are no flybys between 554 Peraga and 20 Massalia because it would be inefficient in terms of both mass and propellant for the spacecraft to travel back to the orbit of Mars or Earth to perform one. Note that additional “forced coasts,” represented by dashed blue lines, appear before and after each flyby. These are 15-day periods

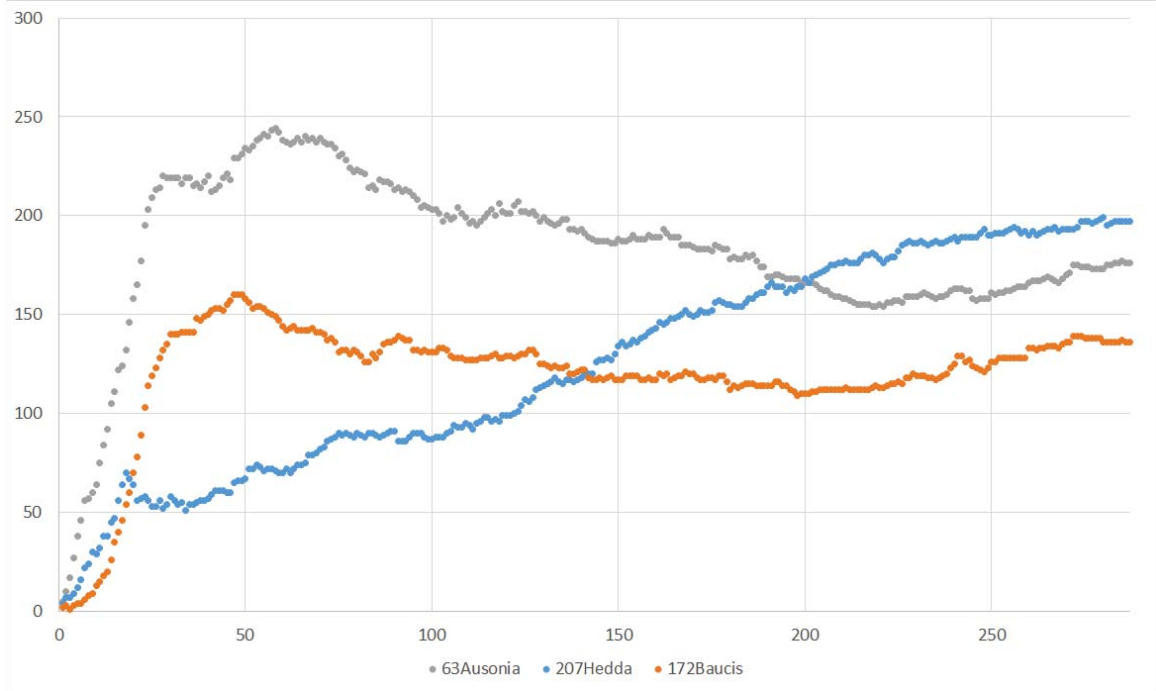


Figure 10: The Rise and Fall of Ausonia, Hedda, and Baucis

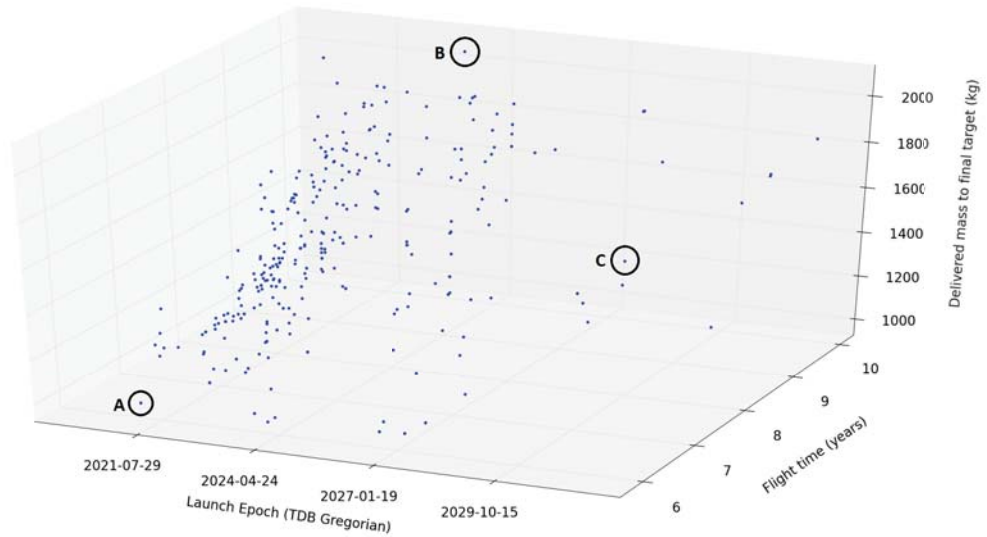


Figure 11: 287th Outer-Loop Generation for the Multi Main-Belt Asteroid Tour Problem, Showing Only Missions Which Deliver at Least 1000 kg in No More Than 10 Years

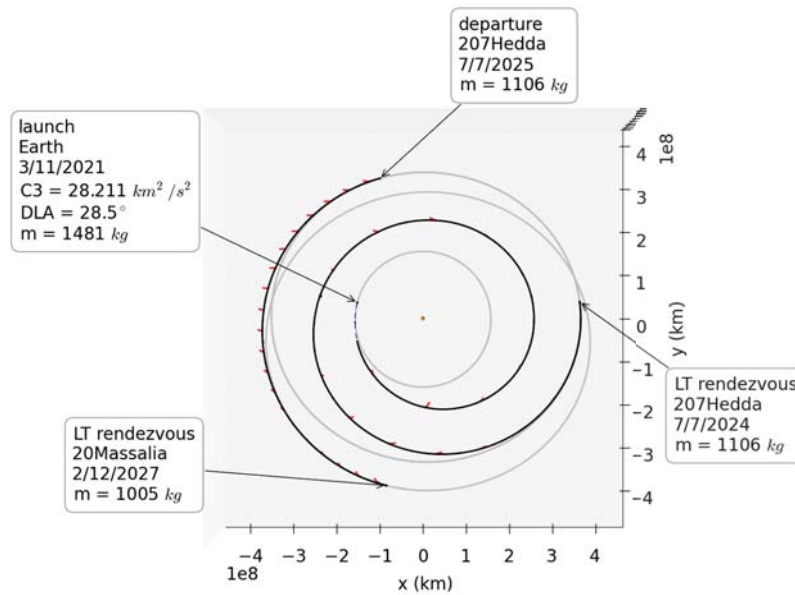


Figure 12: Example Mission A, Delivering 1005 kg to 207 Hedda and 20 Massalia

without thrusting that are enforced so that the navigation team can properly evaluate the effect of the flyby.

Example mission (C), shown in Figure 14 is an example of the popular combination of 172 Baucis and 63 Ausonia in the same mission. Example mission (C) was also chosen because it launches late in the decade of interest, showing the flexibility of the double main-belt rendezvous mission to schedule delays. In this case the spacecraft performs no planetary flybys but takes advantage of the very similar orbits of 172 Baucis and 63 Ausonia. In addition to being nearly circular and having similar semi-major axis values, the two asteroid orbits are nearly coplanar. That is why the combination is so successful in the evolutionary optimization of the mission sequence.

The example missions here are just three of hundreds of choices. They were chosen for this paper because they span the solution space but the final choice of mission is always at the discretion of the customer.

CONCLUSION

Summary

In this work we show that the low-thrust interplanetary mission design problem may be posed as a multi-objective HOCF and efficiently explored via the powerful combination of a multi-objective discrete NSGA-II outer-loop with a MBH+NLP inner-loop. The trade space for a given mission and spacecraft design may be characterized even when there is only enough time to evaluate a fraction of the full combinatorial design problem. A companion paper by these authors extends the technique described here to trade spaces that include not only trajectory variables but also propulsion and power systems trades [36].

The algorithm described here has revolutionized the low-thrust interplanetary mission design process at NASA Goddard Space Flight Center for three reasons. First, multiple mission cases may now be studied simultaneously, limited only by available computing power. Second, mission design engineers can now spend more time with the customer and with spacecraft hardware engineers so that they can fully understand the scientific and engineering context of their work and deliver better value to the customer. Third, good mission ideas are much less likely to be rejected due to lack of time to work on mission design, and bad ideas are much more likely to be rejected before they consume too many resources because the full space of “what if” options may be explored autonomously before a large team of specialists is brought to bear.

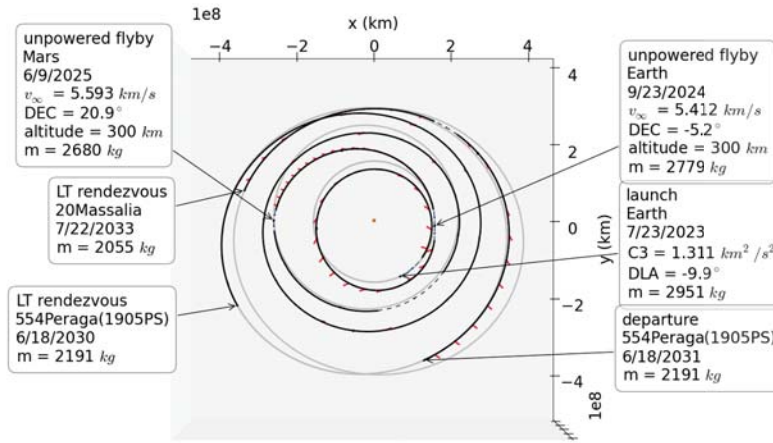


Figure 13: Example Mission B, Delivering 2055 kg to 554 Peraga and 20 Massalia

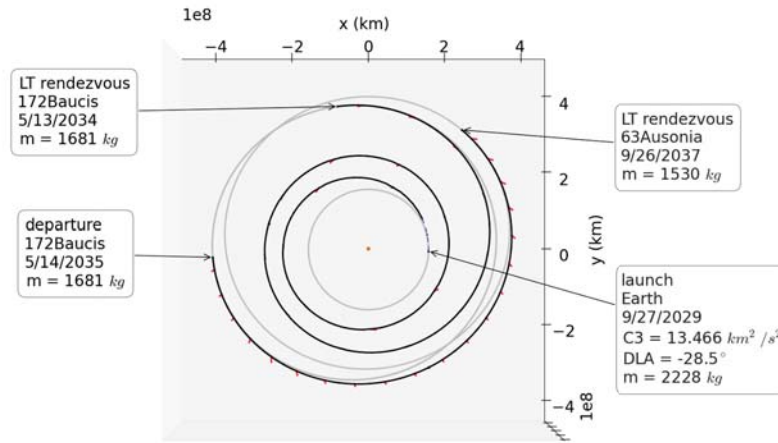


Figure 14: Example Mission C, Delivering 1530 kg to 172 Baucis and 63 Ausonia

Skilled analysts are expensive. With the multi-objective HOCF automaton described in this work, analysts can focus on understanding the customer's needs and the spacecraft's capabilities and also detailed design work, leaving repetitive tasks to the computer. In the long term, this should lead to better mission proposals and therefore better missions.

The algorithms described in this work are available as part of the open-source EMTG project [37]. The authors encourage the reader to examine and use them and welcome opportunities for collaboration.

Future Work

There are two areas of future work to improve the algorithm presented here: improvements to the inner-loop and improvements to the outer-loop. For the outer-loop, no systematic exploration of NSGA-II's tuning parameters has been conducted. In addition, it may be beneficial to encode the outer-loop using binary values instead of integers because binary encoding may enable the GA to more readily innovate, i.e. add new genes that are not already present in the population. Third, there may be other multi-objective algorithms appropriate to this problem.

It may be possible to choose a better transcription for the inner-loop. The Sims-Flanagan transcription is considered to be one of the most robust low-thrust transcriptions when an initial guess is supplied but may

not be the most efficient when combined with MBH. It is merely the only - to these authors' knowledge - low thrust transcription that has been extensively tested with MBH [22, 24, 38]. Other transcriptions, such as collocation methods [39] or direct parallel shooting [40] should be explored.

Work is under-way to extend the multi-objective HOC automaton to space mission design using chemical propulsion. The inner-loop problem for chemical missions is well-studied [41, 42] and some single-objective HOC [11] and mixed-integer [13, 14] approaches have been published, but the full multi-objective HOC has not been done.

ACKNOWLEDGMENTS

The authors would like to acknowledge Donald Ellison and Professor Bruce Conway at the University of Illinois for their ongoing contributions to the algorithms behind EMTG and HOC in general. Frank Vaughn and Wayne Yu of NASA Goddard Space Flight Center and Ryne Beeson of the University of Illinois assisted with testing. This research was an intermediate product of several mission design studies at NASA Goddard Space Flight Center and by the Satellite Servicing Capabilities Office (SSCO).

REFERENCES

- [1] J. Longuski and S. Williams, "Automated design of gravity-assist trajectories to Mars and the outer planets," *Celestial Mechanics and Dynamical Astronomy*, Vol. 52, No. 3, 1991, pp. 207–220.
- [2] A. Petropoulos and J. Longuski, "Shape-based algorithm for automated design of low-thrust, gravity-assist trajectories," *Journal of Spacecraft and Rockets*, Vol. 41, September-October 2004, pp. 787–796.
- [3] B. Wall and B. Conway, "Shape-Based Approach to Low-Thrust Rendezvous Trajectory Design," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, 2009, pp. 95–101.
- [4] B. Wall and D. Novak, "A 3D Shape-Based Approximation Method for Low-Thrust Trajectory Design," *AAS Astrodynamics Specialist Conference*, Girdwood, AK, July 2011.
- [5] J. Downing, "Pyxis Tool," Tech. Rep. NASA/TM-2006-214139, Flight Dynamics Branch (FDAB), NASA Goddard Space Flight Center, August 2006.
- [6] J. Sims, P. Finlayson, E. Rinderle, M. Vavrina, and T. Kowalkowski, "Implementation of a low-thrust trajectory optimization algorithm for preliminary design," *AIAA/AAS Astrodynamics Specialist Conference*, August 2006.
- [7] T. T. McConaghy, *GALLOP Version 4.5 User's Guide*. School of Aeronautics and Astronautics, Purdue University, 2005.
- [8] M. Buss, M. Glocker, M. Hardt, O. v. Stryk, R. Bulirsch, and G. Schmidt, "Nonlinear Hybrid Dynamical Systems: Modeling, Optimal Control, and Applications," *Lecture Notes in Control and Information Sciences*, Vol. 279, 2002, pp. 311–335.
- [9] I. M. Ross and C. D'Souza, "Hybrid Optimal Control Framework for Mission Planning," *Journal of Guidance, Control, and Dynamics*, Vol. 28, July-August 2005, pp. 686–697.
- [10] B. Conway, C. Chilan, and B. Wall, "Evolutionary principles applied to mission planning problems," *Celestial Mechanics and Dynamical Astronomy*, Vol. 97, No. 2, 2007, pp. 73 – 86.
- [11] J. Englander, B. Conway, and T. Williams, "Automated Mission Planning via Evolutionary Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 6, 2012, pp. 1878–1887.
- [12] J. A. Englander, *Automated Trajectory Planning for Multiple-Flyby Interplanetary Missions*. PhD thesis, University of Illinois at Urbana-Champaign, April 2013.
- [13] A. Gad and O. Abdelkhalik, "Hidden Genes Genetic Algorithm for Multi-Gravity-Assist Trajectories Optimization," *Journal of Spacecraft and Rockets*, Vol. 48, July-August 2011, pp. 629–641.
- [14] O. Abdelkhalik and A. Gad, "Dynamic-Size Multi-Population Genetic Optimization for Multi-Gravity-Assist Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 35, March-April 2012, p. 520529.
- [15] M. Vasile and S. Campagnola, "Design of low-thrust multi-gravity assist trajectories to Europa," *Journal of the British Interplanetary Society*, Vol. 62, No. 1, 2009, pp. 15–31.
- [16] V. Coverstone-Carroll, J. Hartmann, and W. Mason, "Optimal multi-objective low-thrust spacecraft trajectories," *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, No. 24, 2000, pp. 387 – 402.
- [17] M. Vavrina and K. Howell, "Multiobjective Optimization of Low-Thrust Trajectories Using a Genetic Algorithm Hybrid," *AAS/AIAA Space Flight Mechanics Meeting*, February 2009. AAS 09-141.

- [18] M. Vasile and F. Zuiani, "MACS: A hybrid multiobjective optimization algorithm applied to space trajectory optimization, *Journal of Aerospace Engineering*," *Proceedings of the Institution of Mechanical Engineers Part G*, Vol. 225, 2011, pp. 1211–1227.
- [19] K. Deb, S. Agrawal, A. Pratap, , and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, April 2002, pp. 182–197.
- [20] J. A. Sims and S. N. Flanagan, "Preliminary Design of Low-Thrust Interplanetary Missions," *AAS/AIAA Astrodynamics Specialist Conference*, Girdwood, Alaska, August 1999.
- [21] R. Leary, "Global optimization on funneling landscapes," *Journal of Global Optimization*, Vol. 18, December 2000, pp. 367–383.
- [22] C. Yam, D. d. Lorenzo, and D. Izzo, "Low-Thrust Trajectory Design as a Constrained Global Optimization Problem," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 225, 2011, pp. 1243–1251.
- [23] J. A. Englander and A. C. Englander, "Tuning Monotonic Basin Hopping: Improving the Efficiency of Stochastic Search as Applied to Low-Thrust Trajectory Optimization," *24th International Symposium on Space Flight Dynamics, Laurel, MD*, May 2014.
- [24] "PaGMO (Parallel Global Multiobjective Optimizer)," <http://pagmo.sourceforge.net/pagmo/index.html>.
- [25] J. Prussing and B. Conway, *Orbital Mechanics*. Oxford University Press, New York, 1993.
- [26] B. A. Conway, "An Improved Method due to Laguerre for the Solution of Kepler's Equation," *Celestial Mechanics*, Vol. 39, No. 2, 1986, pp. 199–211.
- [27] J. A. Englander, D. H. Ellison, and B. A. Conway, "Global Optimization of Low-Thrust, Multiple-Flyby Trajectories and Medium and Medium-High Fidelity," *AAS/AIAA Space Flight Mechanics Meeting, Santa Fe, NM*, January 26 - 30 2014.
- [28] "SPICE Ephemeris," August 2013. <http://naif.jpl.nasa.gov/naif/>.
- [29] J. A. Englander, B. A. Conway, and T. Williams, "Automated Interplanetary Mission Planning," *AAS/AIAA Astrodynamics Specialist Conference, Minneapolis, MN*, August 2012.
- [30] V. Pareto, *Manuale di Economica Politica*. Milano, Italy: Societa Editrice Libreria, 1906. translated into English by A. Schwier as *Manual of Political Economy*, MacMillan Press, New York, 1971.
- [31] C. A. Coello, G. B. Lamont, and D. A. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*, 2nd ed. Springer-Verlag, 2006.
- [32] "MPI: A Message-Passing Interface Standard," tech. rep., Message Passing Forum, Knoxville, TN, USA, 1994.
- [33] A. Cassioli, D. Izzo, D. Di Lorenzo, M. Locatelli, and F. Schoen, "Global Optimization Approaches for Optimal Trajectory Planning," *Modeling and Optimization in Space Engineering* (G. Fasano and J. D. Pintr, eds.), Vol. 73 of *Springer Optimization and Its Applications*, pp. 111–140, Springer New York, 2013.
- [34] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Rev.*, Vol. 47, No. 1, 2005, pp. 99–131.
- [35] D. Oh, S. Benson, K. Witzberger, and M. Cupples, "Performance evaluation of an expended range XIPS ion thruster system for NASA science missions," *40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Ft. Lauderdale, FL*, July 2004.
- [36] M. Vavrina, J. Englander, and A. Ghosh, "Coupled Low-Thrust Trajectory and Systems Optimization Via Multi-objective Hybrid Optimal Control," *AAS/AIAA Space Flight Mechanics Meeting*, January 2015. AAS 15-397.
- [37] "EMTG (Evolutionary Mission Trajectory Generator)," <https://sourceforge.net/projects/emtg/>.
- [38] J. A. Englander, D. H. Ellison, and B. A. Conway, "Global Optimization of Low-Thrust, Multiple-Flyby Trajectories at Medium and Medium-High Fidelity," *AAS/AIAA Space-Flight Mechanics Meeting, Santa Fe, NM*, January 2014.
- [39] P. Enright and B. Conway, "Discrete Approximations to Optimal Trajectories Using Collocation and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 994 – 1002.
- [40] P. Enright and B. Conway, "Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 15, July-August 1992, pp. 994–1002.
- [41] T. Vinkó and D. Izzo, "Global Optimisation Heuristics and Test Problems for Preliminary Spacecraft Trajectory Design," Tech. Rep. GOHTPPSTD, European Space Agency, the Advanced Concepts Team, 2008.
- [42] M. Vasile, E. Minisci, and M. Locatelli, "Analysis of Some Global Optimization Algorithms for Space Trajectory Design," *Journal of Spacecraft and Rockets*, Vol. 47, March-April 2010, pp. 334–344.